



# TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	09/625,518
Filing Date	July 25, 2000
First Named Inventor	Craig David Weissman
Art Unit	2172
Examiner Name	Gewn Liang
Attorney Docket Number	EPI-015 US (7008042001)
Total Number of Pages in This Submission	22

## ENCLOSURES (check all that apply)

<input checked="" type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment / Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Reply to Missing Parts/Incomplete Application <input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) ____ <input type="checkbox"/> Landscape Table on CD	<input type="checkbox"/> After Allowance Communication to TC <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input checked="" type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): <b>Express Mail Label No.: EV347036200US</b> <b>Return Postcard</b>
<b>Remarks</b>		

## SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm	Bingham McCutchen LLP		
Signature			
Printed Name	Saina S. Shamilov		
Date	August 5, 2005	Reg. No.	48,266

## CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this paper is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" (Express Mail Label No. EV 347 036 200 US under 37 CFR § 1.10 on the date indicated and is addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Signature			
Typed or printed name	Sheila Badon	Date	August 5, 2005

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

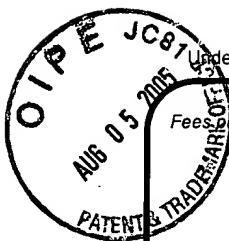
08-08-05

PTO/SB/17 (12-04v2)

Approved for use through 07/31/2006. OMB 0651-0032

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



Effective on 12/08/2004.  
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

# FEE TRANSMITTAL for FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT (\$)**500.00**

## Complete if Known

Application Number 09/625,518  
Filing Date July 25, 2000  
First Named Inventor Craig David Weissman  
Examiner Name Gwen Liang  
Art Unit 2172  
Attorney Docket No. EPI-015 US (7008042001)

## METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify) : \_\_\_\_\_  
☒ Deposit Account Deposit Account Number: 502518 Deposit Account Name: Bingham McCutchen LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, except for the filing fee  
☒ Charge any additional fee(s) or underpayments of fee(s) ☐ Credit any overpayments  
Under 37 CFR 1.16 and 1.17

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

## FEE CALCULATION

### 1. BASIC FILING, SEARCH, AND EXAMINATION FEES

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Fee (\$)	Small Entity Fee(\$)	Fee(\$)	Small Entity Fee(\$)	Fee(\$)	Small Entity Fee(\$)	
Utility	300	150	500	250	200	100	_____
Design	200	100	100	50	130	65	_____
Plant	200	100	300	150	160	80	_____
Reissue	300	150	500	250	600	300	_____
Provisional	200	100	0	0	0	0	_____

### 2. EXCESS CLAIM FEES

#### Fee Description

Each claim over 20 (including Reissues)

Fee (\$) 50

Small Entity Fee (\$) 25

Each independent claim over 30 (including Reissues)

200

100

Multiple dependent claims

360

180

#### Total Claims

#### Extra Claims

#### Fee(\$)

#### Fee Paid (\$)

#### Multiple Dependent Claims

\_\_\_\_\_ -20 or HP= \_\_\_\_\_ x \_\_\_\_\_ = \_\_\_\_\_

Fee (\$) Fee Paid (\$)

HP = highest number of total claims paid for, if greater than 20.

#### Indep. Claims

#### Extra Claims

#### Fee(\$)

#### Fee Paid (\$)

\_\_\_\_\_ - 3 or HP= \_\_\_\_\_ x \_\_\_\_\_ = \_\_\_\_\_

HP = highest number of independent claims paid for, if greater than 3.

### 3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets Extra Sheets Number of each additional 50 or fraction thereof Fee (\$) Fee Paid (\$)  
\_\_\_\_\_ - 100 = \_\_\_\_\_ / 50 = \_\_\_\_\_ (round up to a whole number) x \_\_\_\_\_ = \_\_\_\_\_

### 4. OTHER FEE(S)

Non-English Specification, \$130 fee (no small entity discount)

Other (e.g., late filing surcharge): Reply Brief Under 37 CFR § 1.193

#### Fees Paid (\$)

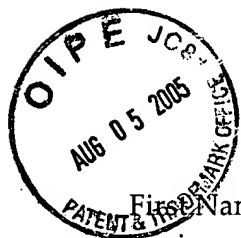
500.00

## SUBMITTED BY

Signature		Registration No. (Attorney/Agent)	48,266	Telephone	650-849-4988
Name (Print/Type)	Saina S. Shamilo	Date	August 5, 2005		

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing this form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

First Named Inventor: Craig David Weissman  
Assignee: E.piphany, Inc.  
Filing Date: July 25, 2000  
Application No.: 09/625,518  
Title: METHOD AND APPARATUS FOR CREATING A WELL-FORMED DATABASE  
SYSTEM USING A COMPUTER /  
Examiner: Gwen Liang  
Group Art Unit: 2172  
Docket No.: EPI-015 US (7008042001)

---

MAIL STOP APPEAL BRIEF - PATENTS

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

**REPLY BRIEF UNDER 37 CFR § 1.193**

The Assignee submit this Reply Brief pursuant to 37 CFR 1.193 in response to the Examiner's Supplemental Answer mailed on June 6, 2005. This brief replaces the reply brief that was submitted on June 17, 2004. This brief is submitted in triplicate.

08/08/2005 SDENBOB1 00000080 502518 09625518

01 FC:1402 500.00 DA

REMARKS

The Assignees respectfully submit that the Examiner's Answer, including the response to arguments, is a copy of the final rejection, which fails to rebut our response in the Appeal Brief, fails to address each of the elements of claim 1, and fails to present any new arguments.

For example, on page 17 of the Answer, in the Response to argument portion of the Answer, the Examiner writes that "Feuche combined into Smiley clearly teach an automated process of creating a set of database tables using the definitions as claimed in the applicant's invention." The Examiner then cites the final Office action. Thus, the Examiner's Answer merely reiterates its final Office Action and adds nothing new to rebut the Assignee's Brief.

The Examiner then expresses, on pages 18 and 19 of the Answer, multiple legal conclusions about the obviousness of claim 1 based on Smiley in view of Feuche. The Examiner again fails to address each element of claim 1. Instead, the Examiner addresses features that are not recited in claim 1.

Claim 1 recites:

A method of creating a system for creating a well-formed database system using a computer, the method comprising:

the computer accessing a definition of the system, the definition defining a schema for use by the system, the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns; and

the computer using the definition to generate the set of tables.

The Examiner's legal argument on obviousness combination of Smiley and Feuche are now addressed.

The Examiner writes that "It is obvious that the link's DB2 utilities are run from a computer because it is used as an interface to create DB2 tables from definitions created in Excelerator." The Examiner's statement, however, does not show that Feuche discloses what is recited in Claim 1. Just because the DB2 tables can be created from definitions does not teach or suggest that the system disclosed in Feusche is capable of generating a set of tables using a definition **defining a set of relationships between the tables**, as claimed in the present application. In fact, Feusche fails to disclose or suggest that the definitions used by the link include a set of relationships between the tables. In fact, Feusche lists the definitions on page 2 and fails to mention any relationship between tables.

The Examiner writes that:

"It is also obvious that when the link's DB2 utilities are run from a computer, the computer then accessing and using the logical definitions to generate the set of tables. Although it should be obvious to one having ordinary skill in the art at the time the invention was made to recognize that the logical definitions taught in Feuche would include the same types of logical definitions as claimed in the applicant's invention, the Feuche reference does not disclose the logical definitions in the level of detail as claimed in the applicant's invention."

The Assignees respectfully submit that the Examiner has no factual basis and in fact fails to cite to any portions of Smiley or Feusche to support the conclusion that "it should be obvious ... that the logical definitions taught in Feuche would include the same types of logical definitions as claimed in the applicant's invention...." Indeed, there are more.

In fact, the Feuce article merely discloses an interface "to transform logical design data produced in Execlerator into physical design data -- specifically, into structured query language (SQL)...." (Feuche page 1). Therefore, Feuche does not disclose "the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables," as recited in claim 1. Feuche further neither discloses nor suggests "the computer using the definition to generate the set of tables," the definition "defining a schema for use by the system, the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns," as recited in claim 1.

Generating a set of tables based on "the schema defining . . . a set of relationships between the tables of the set of tables" is neither as trivial nor obvious in light of Feusche as the Examiner suggests. Related tables are usually related to each other through a common field. *See Database And Knowledge - Base Systems* at pp. 49-50 (relevant pages attached herewith). For example, the following two tables are related through their DNAME field:

DEPTS TABLE

DNAME	DEPT#
Produce	12
Cheese	31
Meat	5

MANAGES

ENAME	DNAME
Esther Eggplant	Produce
Larry Limburger	Cheese
Harry Hamhock	Meat

Relational tables may be combined into a single table to save storage space and to reduce query processing time. *See Database And Knowledge - Base Systems* at pp. 49-50 (relevant pages attached herewith). For example, the above two tables may be combined into a single table with less entries:

DNAME	DEPT#	MGR
Produce	12	Esther Eggplant
Cheese	31	Larry Limburger
Meat	5	Harry Hamhock

In order to generate related tables the relationship information must be provided to the system that in turn has a capability to generate the relational tables using the relationship information. Feusche does not even suggest that the Excelerator's dictionary includes any information about relationships between tables that may be used by a computer when generating a set of tables. Nor does Feusche suggest that the link has a capability to generate tables using any relational information.

The Examiner writes that the:

“Smiley reference, however, fully disclose those definitions, especially definitions that define a set of relationships between the tables (See for example: col. 3 lines 8-11, col. 3 lines 27-44, col. 4 lines 25-32) and programs that operate on the set of tables and the set of table columns (See for example: col. 6 line 66 - col. 7 line 3). In conclusion, the Smiley reference teaching the definition as recited in claim 1....”

First, the Assignees respectfully submit that the Examiner mischaracterizes claim 1. In fact, claim 1 recites “the definition defining a schema for use by the system, the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table column.”

The Assignees also respectfully submit that Smiley, which discloses a method for a user to formulate relationships between data objects, does not disclose “the definition defining a schema for use by the system, the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table column” as recited in claim 1.

For example, Smiley at column 3 lines 8-11 discloses:

Alternatively, the attributes of OBJECT 12 may be configured as other OBJECTS, which are logically related to OBJECT 12 via a RELATIONSHIP entity 14.



Smiley at column 3 lines 27-33 discloses:

RELATIONSHIP entity 14 preferably contains attributes or fields which record the name of RELATIONSHIP entity 14 represented by characters, the names or identifiers, and types of OBJECTS 12 between which this relationship holds, a sequence number to ensure the uniqueness of RELATIONSHIP entity 14, and the name of a METHOD entity 16 which implements RELATIONSHIP 14.

Smiley at column 4 lines 25-32 discloses:

Another OBJECT 12 is ENTITY 21, which is a logical collection of ATTRIBUTES 20. ENTITY 21 CONTAINS 22 ATTRIBUTE 20. CONTAINS relationship 22 documents the one-to-many relationship that each ENTITY 21 has with the attributes it contains. This relationship along with the ATTRIBUTE name, serves to uniquely identify each ATTRIBUTE.

The final passage from Smiley cited by the Examiner on page 19 of the Answer is column 6 line 65 through column 7 line 3, which discloses:

The OBJECTS for an operational system's information repository preferably include ATTRIBUTES (not shown), which contain the data of interest to the enterprise. OPERATIONAL METHODS are the existing application programs that display or update the operational data.

None of these passages teaches or even suggests a computer generating a set of tables based on a definition defining a set of relationships between the tables, as claimed in the present application.

Therefore, the Assignees submit that Smiley does not disclose "the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables," as recited in claim 1. The Assignees respectfully submit that Smiley further neither discloses nor suggests "the computer using the definition to generate the

set of tables,” the definition “defining a schema for use by the system, the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns” as recited in claim 1.

Even if Smiley and Feuche were combined as suggested by the Examiner, the combination would neither teach nor suggest “the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables,” as recited in claim 1. The combination would further neither teach nor suggest “the computer using the definition to generate the set of tables,” the definition “defining a schema for use by the system, the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns” as recited in claim 1.

Therefore, the Assignees submit that claim 1 is patentable under 35 U.S.C. §103 based on Smiley in view of Feuche.

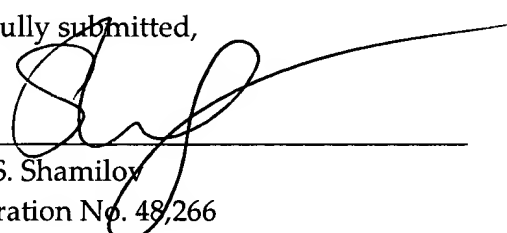
CONCLUSION

For the above reasons, the Assignees respectfully submit that the rejection of claims 1-47 based on 35 U.S.C. § 103(a) has been overcome. Accordingly, the Assignees request that the Board of Patent Appeals and Interferences overrule the Examiner and allow claims 1-47.

DATE: August 5, 2005

Respectfully submitted,

By: \_\_\_\_\_

  
Saina S. Shamilov  
Registration No. 48,266

**Bingham McCutchen LLP**  
Three Embarcadero Center, Suite 1800  
San Francisco, California 94111  
Telephone: (650) 849-4400  
Telefax: (650) 849-4800

Listing of Claims:

1. (Original) A method of creating a system for creating a well-formed database system using a computer, the method comprising:  
  
the computer accessing a definition of the system, the definition defining a schema for use by the system, the schema defining a set of tables, a set of columns that correspond to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns; and  
  
the computer using the definition to generate the set of tables.
2. (Original) The method of claim 1 wherein the set of tables includes a first table and a second table, wherein the first table includes a first column, wherein the second table includes a second column, and wherein the first column and the second column are related by a join and are therefore guaranteed to be from the same domain.
3. (Original) The method of claim 1 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to one relationship, and wherein the generating the set of tables includes automatically generating a foreign key column in the first table, wherein the foreign key column is for holding a foreign key to the second table.

4. (Original) The method of claim 1 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to many relationship, and wherein the generating the set of tables includes automatically generating an associative table corresponding to the first table and the second table, and wherein the associative table has a unique value created for each unique many-to-many relationship between the first table and the second table.

5. (Original) The method of claim 1 wherein the set of tables includes a first table and a second table, and wherein the first table includes one or more columns from the second table, and wherein said one or more columns are automatically populated from the one or more columns.

6. (Original) The method of claim 1 wherein the computer using the definition to generate the set of tables also includes the computer performing at least some of the set of operations on at least some of the set of tables.

7. (Original) The method of claim 1 wherein a transaction type column is automatically included in some tables of the set of tables.

8. (Original) The method of claim 1 wherein a date column is automatically included in some tables of the set of tables.

9. (Original) The method of claim 1 wherein a source system key column is automatically included in some tables of the set of tables.

10. (Original) The method of claim 1 wherein the definition defines a set of source system extraction operations, wherein the set of source system extraction operations are for extracting data from a source system and for manipulating the data for populating the database, and wherein the set of source system extraction operations correspond to the schema definition.

11. (Original) The method of claim 10 wherein the source system extraction operations correspond to the schema definition by populating source system data into the database system according the schema definition.

12. (Original) The method of claim 1 wherein the definition defines a set of aggregates for the database system, the set of aggregates corresponding to the schema definition, the method further comprising:

the computer using the definition to create a set of aggregate tables corresponding to the set of aggregates; and  
populating the set of aggregate tables.

13. (Original) The method of claim 12 wherein the set of aggregates corresponds to the schema definition by defining which aggregates should be made from which tables in the database system.

14. (Original) The method of claim 12 wherein the definition defines an aggregate operation for an aggregate of the set of aggregates.

15. (Original) The method of claim 14 wherein the aggregate operation includes a SUM operation.

16. (Original) The method of claim 14 wherein the aggregate operation includes an AVERAGE operation.

17. (Original) The method of claim 1 wherein the definition includes a user interface definition for querying the database and for presenting results, the user interface definition corresponding to the schema definition.

18. (Original) The method of claim 17 wherein the user interface definition specifies which columns from which tables can be used in a query.

19. (Original) The method of claim 1 wherein the definition defines a set of source system extraction operations, a set of aggregates, and a user interface definition, that correspond to the schema definition.

20. (Original) The method of claim 1 wherein the database system includes a datamart, wherein the schema definition includes a star schema definition, wherein the set of tables includes a set of fact tables and a set of dimension tables.

21. (Original) A system comprising:

a database system;

a first program for accessing a definition of the schema for the database system, the

schema defining a set of tables, a set of columns corresponding to the set of tables,

and a set of relationships between the tables of the set of tables, the definition

further defining a set of operations for manipulating the data, the set of operations

defining programs that operate on the set of tables and the set of table columns, the

first program further for using the definition to generate the set of tables.

22. (Original) The system of claim 21 wherein the set of tables includes a first table and a second table, wherein the first table includes a first column, wherein the second table includes a second column, and wherein the first column and the second column are related by a join and are therefore guaranteed to be from the same domain.

23. (Original) The system of claim 21 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to one relationship, and wherein the generating the set of tables includes automatically generating a foreign key column in the first table, wherein the foreign key column is for holding a foreign key to the second table.

24. (Original) The system of claim 21 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to many relationship, and wherein the generating the set of tables includes automatically generating an associative table corresponding to the first table and the second table, and wherein the associative table has a unique value created for each unique many-to-many relationship between the first table and the second table.

25. (Original) The system of claim 21 wherein the set of tables includes a first table and a second table, and wherein the first table includes one or more columns from the second table, and wherein said one or more columns are automatically populated from the one or more columns.



26. (Original) The system of claim 21 wherein the first program includes an enterprise manager for accessing the definition, causing the generation of the set of tables, and causing the population of the tables.

27. (Original) The system of claim 21 further comprising a database, the database for storing the set of tables.

28. (Original) The system of claim 21 further comprising an aggregate building program for accessing a definition of a set of aggregates and the definition of the schema and for generating the set of aggregates from the definition of the set of aggregates and the definition of the schema.

29. (Original) The system of claim 21 further comprising a query and reporting program for generating a user interface from a definition of the user interface and the definition of the schema.

30. (Original) A system comprising:  
means for accessing a definition of the system, the definition defining a schema for use by the system, the schema defining a set of tables, a set of columns corresponding to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns; and  
means for using the definition to generate the set of tables.

31. (Original) The system of claim 30 wherein the set of tables includes a first table and a second table, wherein the first table includes a first column, wherein the second table includes a second column, and wherein the first column and the second column are related by a join and are therefore guaranteed to be from the same domain.

32. (Original) The system of claim 30 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to one relationship, and wherein the generating the set of tables includes automatically generating a foreign key column in the first table, wherein the foreign key column is for holding a foreign key to the second table.

33. (Original) The system of claim 30 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to many relationship, and wherein the generating the set of tables includes automatically generating an associative table corresponding to the first table and the second table, and wherein the associative table has a unique value created for each unique many-to-many relationship between the first table and the second table.

34. (Original) The system of claim 30 wherein the set of tables includes a first table and a second table, and wherein the first table includes one or more columns from the second table, and wherein said one or more columns are automatically populated from the one or more columns.

35. (Original) The system of claim 34 wherein the definition of the system further includes a definition of the aggregates for the system, the system further comprising:

means for generating a set of aggregates from the definition of the aggregates and the definition of the schema.

36. (Currently Amended) The system of claim 33 wherein the definition of the system further includes a definition of ~~the~~ a user interface for the system, the system further comprising:

means for generating the user interface from the definition of the user interface and the definition of the schema.

37. (Original) The system of claim 33 wherein the definition of the system includes a definition of aggregates for use in the system and a definition of a query and reporting mechanism interface for the system, the set of tables includes a set of fact tables and a set of dimension tables, and wherein the system further comprises:

means for generating the set of fact tables;

means for generating the set of dimension tables;

means for generating a set of aggregate tables; and

means for generating a query and reporting mechanism interface.

38. (Original) A computer program product comprising:

a memory medium; and

a computer program stored on the memory medium, the computer program comprising

instructions for accessing a definition of a system, the definition defining a schema for use by the system, the schema defining a set of tables, a set of columns corresponding to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns, and instructions for using the definition to generate the set of tables.

39. (Original) The computer program product of claim 38 wherein the set of tables includes a first table and a second table, wherein the first table includes a first column, wherein the second table includes a second column, and wherein the first column and the second column are related by a join and are therefore guaranteed to be from the same domain.

40. (Original) The computer program product of claim 38 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to one relationship, and wherein the generating the set of tables includes automatically generating a foreign key column in the first table, wherein the foreign key column is for holding a foreign key to the second table.

41. (Original) The computer program product of claim 38 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to many relationship, and wherein the generating the set of tables includes automatically generating an associative table corresponding to the first table

and the second table, and wherein the associative table has a unique value created for each unique many-to-many relationship between the first table and the second table.

42. (Original) The computer program product of claim 38 wherein the set of tables includes a first table and a second table, and wherein the first table includes one or more columns from the second table, and wherein said one or more columns are automatically populated from the one or more columns.

43. (Original) A computer data signal embodied in a carrier wave comprising:  
a computer program, the computer program comprising instructions for accessing a definition of a system, the definition defining a schema for use by the system, the schema defining a set of tables, a set of columns corresponding to the set of tables, and a set of relationships between the tables of the set of tables, the definition further defining a set of operations for manipulating the data, the set of operations defining programs that operate on the set of tables and the set of table columns, and instructions for using the definition to generate the set of tables.

44. (Original) The computer data signal embodied in the carrier wave of claim 43 wherein the set of tables includes a first table and a second table, wherein the first table includes a first column, wherein the second table includes a second column, and wherein the first column and the second column are related by a join and are therefore guaranteed to be from the same domain.

45. (Original) The computer data signal embodied in the carrier wave of claim 43 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to one relationship, and wherein the generating the set of tables includes automatically generating a foreign key column in the first table, wherein the foreign key column is for holding a foreign key to the second table.

46. (Original) The computer data signal embodied in the carrier wave of claim 43 wherein the set of tables includes a first table and a second table, and wherein the definition defines that the first table relates to the second table by a many to many relationship, and wherein the generating the set of tables includes automatically generating an associative table corresponding to the first table and the second table, and wherein the associative table has a unique value created for each unique many-to-many relationship between the first table and the second table.

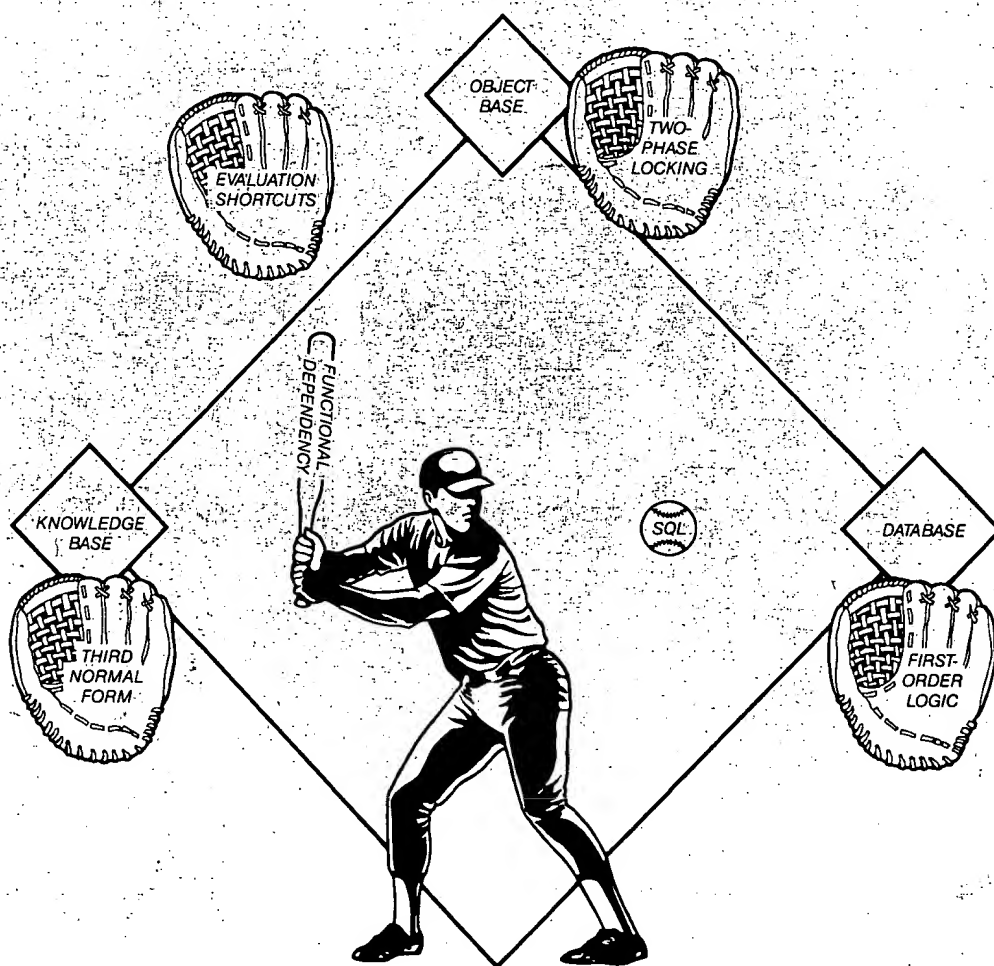
47. (Original) The computer data signal embodied in the carrier wave of claim 43 wherein the set of tables includes a first table and a second table, and wherein the first table includes one or more columns from the second table, and wherein said one or more columns are automatically populated from the one or more columns.

BEST AVAILABLE COPY

# PRINCIPLES OF DATABASE AND KNOWLEDGE-BASE SYSTEMS

VOLUME I

Jeffrey D. Ullman



BEST AVAILABLE COPY

PRINCIPLES OF  
DATABASE AND  
KNOWLEDGE - BASE  
SYSTEMS  
VOLUME I

---

Jeffrey D. Ullman  
*STANFORD UNIVERSITY*

COMPUTER SCIENCE PRESS



## 2.3 THE RELATIONAL DATA MODEL

4. If a relation comes from a relationship that is many-one from  $E_1, \dots, E_{k-1}$  to  $E_k$ , then the set of attributes that is the union of the keys for  $E_1, \dots, E_{k-1}$  is normally a key for the relation.

**Example 2.11:** Figure 2.5 lists the thirteen relations from Example 2.9. The set of attributes forming the primary key of each relation is indicated by bold-face letters. Where there is another candidate key, that is indicated by slanted letters. The reader should note the difference between a situation like that in SUPPLIES, where the lone key consists of two attributes, and that of DEPTS, where there are two candidate keys, each consisting of one attribute. The explanation for SUPPLIES is given by rule (4) above, since the relationship supplies is many-one from SUPPLIERS and ITEMS to PRICE, and the first two entity sets have keys SNAME and INAME, respectively. Thus, {SNAME, INAME} forms a key for relation SUPPLIES. The explanation regarding DEPTS is more ad-hoc. We know that DNAME is the key for entity set DEPTS, but we might well decide that DEPT# also should be a key, since the YVCB probably does not intend to give two departments the same number.  $\square$

- (1) EMPS(ENAME, SALARY)
- (2) MANAGERS(ENAME)
- (3) DEPTS(DNAME, *DEPT#*)
- (4) SUPPLIERS(SNAME, SADDR)
- (5) ITEMS(INAME, *ITEM#*)
- (6) ORDERS(O#, DATE)
- (7) CUSTOMERS(CNAME, CADDR, BALANCE)
- (8) WORKS\_IN(ENAME, DNAME)
- (9) MANAGES(ENAME, DNAME)
- (10) CARRIES(INAME, DNAME)
- (11) SUPPLIES(SNAME, INAME, PRICE)
- (12) INCLUDES(O#, INAME, QUANTITY)
- (13) PLACED\_BY(O#, CNAME)

Figure 2.5 Table of relations and keys.

### Relations with Common Keys

When two relations have a candidate key in common, we can combine the attributes of the two relation schemes and replace the two relations by one whose set of attributes is the union of the two sets. One advantage to doing so is that we save the storage space needed to repeat the key values in the two relations. A second is that queries talking about attributes of the two relations can sometimes be answered more quickly if the two relations are combined.

**Example 2.12:** Relations DEPTS and MANAGES from Figure 2.5 each have DNAME as a candidate key; in one case it is the primary key and in the other not. We may thus replace DEPTS and MANAGES by one relation

DEPTS(DNAME, DEPT#, MGR)

Notice that we have decided to call the new relation DEPTS. The attributes DNAME and DEPT# are intended to be the same as the attributes of the same name in the old DEPTS relation, while MGR is intended to be the attribute ENAME from MANAGES. There is nothing wrong with changing the names of attributes, as long as we carry along their intuitive meaning.

In Figure 2.6(a) we see two possible instances for the old relation DEPTS and MANAGES. Figure 2.6(b) shows them combined into one relation, the new DEPTS. Notice that the twelve entries in the two relations have been combined into nine in the single relation, saving a small amount of space. Also, a query like "what is the number of the department that Harry Hamhock manages?" can be answered by consulting the one relation in Figure 2.6(b), while in the database of Figure 2.6(a) we would have to combine the two relations by a possibly expensive operation called the join, discussed in the next section. □

DNAME	DEPT#	ENAME	DNAME
Produce	12	Esther Eggplant	Produce
Cheese	31	Larry Limburger	Cheese
Meat	5	Harry Hamhock	Meat

DEPTS

MANAGES

(a) Old relations.

DNAME	DEPT#	MGR
Produce	12	Esther Eggplant
Cheese	31	Larry Limburger
Meat	5	Harry Hamhock

(b) New relation DEPTS.

**Figure 2.6** Combination of relations with common keys.

## Dangling Tuples

When we combine two or more relations like those in Example 2.12, there is a problem that must be overcome, a problem that if not solved or defined away